Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

Nevillova eliminační metoda

Neville elimination

2013

Kateřina Švédová

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra aplikované matematiky

Zadání bakalářské práce

Student: **Kateřina Švédová**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 1103R031 Výpočetní matematika

Téma: Nevilleova eliminační metoda
Neville elimination

Zásady pro vypracování:

Jednou ze základních úloh lineární algebry je výpočet řešení soustavy lineárních rovnic, který se velmi často provádí pomocí algoritmu Gaussovy eliminační metody. Malou změnou organizace výpočtu vznikne algoritmus Nevilleovy eliminační metody, který má ale poněkud jiné stabilitní vlastnosti. Hlavní náplní této práce bude teoretické a experimentální porovnání obou algoritmů.

Osnova práce:

- soustavy lineárních rovnic
- Gaussova eliminační metoda
- Nevilleova eliminační metoda
- numerické porovnání

Seznam doporučené odborné literatury:

[1] Z. Dostál, V. Vondrák: Lineární algebra. Skriptum VŠB-TU Ostrava, 2011.
[2] P. Alonso, R. Cortina, V. Hernández, J. Ranilla: A study of the performance of Neville elimination using two kinds of partitioning techniques. Linear Algebra and its Applications 332-334(2001), 111-117.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. RNDr. Radek Kučera, Ph.D.**

Datum zadání: 16.11.2012
Datum odevzdání: 07.05.2013

doc. RNDr. Jiří Bouchala, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě dne 6.května 2013

Švédová
....

Ráda bych zde poděkovala vedoucímu bakalářské práce Doc. RNDr. Radku Kučerovi, Ph.D. za jeho ochotu, trpělivost a cenné rady při vzniku této práce. Dále bych ráda poděkovala své rodině a přátelům za morální podporu a v neposlední řadě i svému příteli za pomoc, trpělivost a psychickou podporu.

Abstrakt

Tato práce popisuje přesnost výpočtů prováděných pomocí Gaussovy eliminační metody a Nevillovy eliminační metody. V prvních dvou kapitolách se věnujeme zavedení základních pojmů a jejich definicím. Zavedeme si také pojem matice se známou inverzí. Třetí kapitola se věnuje Gaussově eliminační metodě a čtvrtá kapitola se zabývá Nevillovou eliminační metodou. Obě tyto kapitoly obsahují řešení konkrétního příkladu, popis algoritmu, jeho stručný zápis a zavedení pojmu pivotizace u každé metody.

Klíčová slova

Gaussova eliminační metoda, Nevillova eliminační metoda, Pivotizace, Pivot, Multiplikátor

Abstract

This thesis describes accuracy of the computations, which are performed using Gaussian Elimination Method and Neville Elimination. First two chapters are about introduction of elementary terms and their definitions. There is also introduced term Matrix With Known Inverse. Third chapter is about Gaussian Elimination Method and fourth chapter describes Neville Elimination. Both of these chapters contain solving of an concrete example, description of algorithm, brief notation of this algorithm and introduction of term pivotization for each method.

Keywords

Gaussian Elimination Method, Neville Elimination, Pivoting, Pivot, Multiplier

Použité značení

$\|\cdot\|$	norma
$det\ A$	determinant matice A
H_n	Hilbertova matice řádu n
$\kappa(A)$	číslo podmíněnosti matice A
P_n	Pascalova matice řádu n
$\mathbb{R}$	množina reálných čísel
$\mathbb{R}^{n\times n}$	2D prostor reálných čísel
$\mathcal{V}$	abstraktní vektorový prostor
V_n	Vandermondova matice řádu n

Použité zkratky

GEM	Gaussova eliminační metoda
NEM	Nevillova eliminační metoda

Obsah

Seznam tabulek

Kapitola 1

Úvod

V této práci se zabýváme přesností výpočtů pomocí Gaussovy eliminační metody (GEM) a Nevillovy eliminační metody (NEM). GEM je klasická metoda pro řešení soustav lineárních rovnic. NEM je jí velmi podobná, liší se ale v organizaci výpočtu, což může mít podstatný vliv na přesnost vypočítaných výsledků.
V kapitolách 2 a 3 si zavedeme pojmy jako je soustava lineárních rovnic, matice soustavy, determinant matice, inverzní matice, podmíněnost matice a uvedeme některé matice se známou inverzí. Kapitoly jsou zajištěny studijními materiály [1], [2] a [3].
Nezbytnou částí této práce je popis implementace GEM, GEM s pivotizací, NEM a NEM s pivotizací. V kapitolách 4 a 5 si ukážeme řešený příklad, popis algoritmu a jeho zápis v programovacím jazyce. U každé metody si ukážeme, jak probíhá výběr pivota, výpočet multiplikátoru a získání výsledného vektoru. Literární opory jsou [1], [4] a [5].
V poslední části této práce otestujeme přesnost algoritmů na maticích se známou inverzí. Algoritmy implementované v Matlabu včetně testovacích souborů a získaných výsledků jsou součástí přílohy.

Kapitola 2

Soustavy lineární rovnic s regulární maticí

Soustavy rovnic jsou známým problémem v mnoha různých odvětvích jako např. ve fyzice, v elektrotechnice aj. Soustavy rovnic mohou být lineární, kvadratické, kubické atd. nebo také např. diferenciální. My se zaměříme pouze na soustavy lineárních rovnic s regulární maticí pro vybrané matice se známou inverzí. V této kapitole je čerpáno z [1].

2.1 Soustavy lineárních rovnic

Obecnou soustavou lineárních rovnic myslíme:

$$\begin{array}{ccccccc} a_{11}x_1 & + & \dots & + & a_{1n}x_n & = & b_1 \\ \vdots & & \dots & & \vdots & & \vdots \\ a_{m1}x_1 & + & \dots & + & a_{mn}x_n & = & b_m \end{array} \tag{2.1}$$

Při obecném řešení soustavy lineárních rovnic můžou nastat tyto možnosti:

- Soustava má právě jedno řešení.
- Soustava má nekonečně mnoho řešení.
- Soustava nemá řešení.

2.1.1 Ekvivalentní úpravy

Hlavní a základní myšlenka řešení soustavy lineárních rovnic je nahrazení dané soustavy jinou soustavou, která má stejné řešení, ale získat jej bude jednodušší. Např. když budeme mít soustavu dvou rovnic o dvou neznámých, která obsahuje alespoň jednu rovnici s jednou neznámou, bude tato soustava rovnic výrazně jednodušší, protože můžeme takovou rovnici řešit nezávisle na druhé rovnici. Novou soustavu můžeme získat použitím tzv. ekvivalentních úprav, které zvolíme tak, aby řešení původní soustavy odpovídalo řešení upravené soustavy.

Ekvivalentní úpravy:

(E1) Vzájemná výměna libovolných dvou rovnic soustavy.

(E2) Násobení obou stran některé rovnice soustavy nenulovým číslem.

(E3) Přičtení násobku některé rovnice soustavy k jiné rovnici.

Ekvivaletní úpravy mají specifickou vlastnost, že s jejich pomocí můžeme z upravené soustavy získat zpět původní soustavu aplikováním úprav v opačném pořadí.

Věta 2.1.1. Jsou-li dvě soustavy lineárních rovnic ekvivalentní, potom mají stejné řešení.

2.1.2 Maticový zápis

Při úpravě rovnic je praktičtější používat maticový zápis, tedy vynecháváme opisování neznámých. Soustava rovnic bude v maticovém zápisu vypadat následovně:

$$\left[\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_m \end{array}\right] \tag{2.2}$$

Tuto tabulku nazýváme rozšířená matice soustavy. Levá část tabulky bez posledního sloupce se nazývá matice soustavy, poslední sloupec je pravá strana soustavy. Pokud budeme mluvit pouze o matici soustavy, tak jí budeme nazývat matice.

Ekvivaletním úpravám soustavy rovnic odpovídají operace s řádky rozšířené matice soustavy, které nazýváme **elementární řádkové operace**:

(e1) Vzájemná výměna libovolných dvou řádků.

(e2) Násobení některého řádku nenulovým číslem.

(e3) Přičtení násobku některého řádku k jinému řádku.

Máme-li dvě matice, z nichž jedna vznikla z druhé pomocí elementárních řádkových operací, budeme říkat, že matice jsou řádkově ekvivalentní. Nyní si můžeme větu 2.1.1 vyjádřit pomocí nových pojmů.

Věta 2.1.2. Mají-li dvě soustavy lineárních rovnic řádkově ekvivalentní rozšířené matice, potom mají stejné řešení.

Kapitola 3

Základní pojmy a definice

V této kapitole si zavedeme základní pojmy a definice, které budeme potřebovat k pochopení dalších kapitol. Informace jsou čerpány z [1], [2], [3], [4].

3.1 Matice a její typy

Získání matice ze soustavy rovnic jsme si ukázali v kapitole 2.1.2. Nyní si definujme pojmy, se kterými se budeme setkávat.

Definice 3.1.1. Matice je schéma čísel tzv. prvků matice. Obsahuje obecně m řádků a n sloupců. Tato matice se poté nazývá matice typu $m \times n$.

Řádky a sloupce matice jsou označeny indexy. Jejich kombinace pak představují pozice jednotlivých prvků matice. Obecně prvek v i-tém řádku a j-tém sloupci značíme a_{ij}.

Matice s obecnými prvky:

$$\begin{bmatrix} a_{11} & \cdots & \cdots & \cdots & a_{1n} \\ a_{21} & \ddots & \ddots & \ddots & a_{2n} \\ \vdots & \ddots & a_{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{m1} & \cdots & \cdots & \cdots & a_{mn} \end{bmatrix}$$

3.1.1 Čtvercová matice

Definice 3.1.2. Matici budeme nazývat čtvercovou, pokud je počet řádků a sloupců shodný.

Příklady čtvercových matic:

$$A_1 = \begin{bmatrix} 2 & 3 \\ 1 & 5 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 1 & 5 \\ 2 & 5 & 2 \\ 3 & 7 & 4 \end{bmatrix}$$

3.1.2 Diagonální matice

Definice 3.1.3. Čtvercová matice se nazývá diagonální, pokud má alespoň jeden nenulový prvek v diagonále a všude jinde nuly. Jinak Takovou matici budeme obvykle značit D.

Příklady diagonálních matic:

$$D_1 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 6 \end{bmatrix}, D_2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

3.1.3 Jednotková matice

Definice 3.1.4. Jednotková matice je diagonální matice, která má na diagonále všechny prvky rovny jedné.

Příklad jednotkové matice:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.1.4 Horní trojúhelníková matice

Definice 3.1.5. Matice se bude nazývat horní trojúhelníková matice, pokud pod hlavní diagonálou budou pouze nulové prvky. Takovou matici budeme obvykle značit U.

Příklady matic typu U:

$$U_1 = \begin{bmatrix} 5 & 1 & 3 \\ 0 & -2 & 4 \\ 0 & 0 & 6 \end{bmatrix}, U_2 = \begin{bmatrix} 2 & 3 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

3.1.5 Dolní trojúhelníková matice

Definice 3.1.6. Matice se bude nazývat dolní trojúhelníková matice, pokud nad hlavní diagonálou budou pouze nulové prvky. Takovou matici budeme obvykle značit L.

Příklady matic typu L:

$$L_1 = \begin{bmatrix} 5 & 0 & 0 \\ 2 & -2 & 0 \\ 1 & 3 & 6 \end{bmatrix}, L_2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 5 & 0 & 4 \end{bmatrix}$$

3.1.6 Symetrická matice

Definice 3.1.7. Symetrická matice je souměrná podle hlavní diagonály, tedy její prvky na pozicích a_{ij} a a_{ji} jsou shodné.

Příklad symetrické matice:

$$A = \begin{bmatrix} 5 & 2 & 1 \\ 2 & -2 & 3 \\ 1 & 3 & 6 \end{bmatrix}$$

3.2 Determinant matice

Abychom zjistili, zda je matice regulární nebo singulární, musíme umět spočítat determinant. Pro determinant je nutný pojem zobrazení, který si předem definujeme.

3.2.1 Zobrazení

Zobrazení je předpis, který každému prvku z jedné množiny, přiřadí právě jeden prvek z jiné množiny.

Definice 3.2.1. Zobrazení f se značí $f : \mathcal{A} \longrightarrow \mathcal{B}$, kde $\mathcal{A}, \mathcal{B}$ jsou množiny.

Množině $\mathcal{A}$ se říká definiční obor, obsahuje prvky zvané vzory a značí se jako $\mathcal{D}(f)$, množině $\mathcal{B}$ se říká obor hodnot, obsahuje prvky zvané obrazy a značí se jako $\mathcal{H}(f)$. Jednoznačnost zobrazení je důležitá, znamená, že každý prvek z množiny $\mathcal{A}$ má právě jeden obraz v množině $\mathcal{B}$. Prvky množiny $\mathcal{B}$ ovšem musí mít alespoň jeden vzor.

3.2.2 Determinant

Nyní, když víme co je zobrazení, můžeme si definovat pojem determinant matice:

Definice 3.2.2. V lineární algebře je determinant zobrazení, které přiřadí každé čtvercové matici A číslo, které označujeme $|A|$ nebo $det\ A$.

Determinanty lze počítat několika způsoby, některé možnosti výpočtu naleznete v [1].
Podle determinantu lze určit, zda je matice singulární nebo regulární.

Definice 3.2.3. Čtvercová matice A se nazývá **singulární**, je-li její determinant nulový. Čtvercová matice A se nazývá **regulární**, je-li její determinant nenulový.

3.3 Inverzní matice

Definice 3.3.1. Inverzní matice A^{-1} k dané matici A je taková matice, která po vynásobení s původní matici dá jednotkovou matici. Inverzní matice A^{-1} k matici A je určena jednoznačně.

Pro získání inverzní matice upravíme soustavu $[A|I]$ pomocí ekvivalentních úprav na soustavu $[I|B]$, kde B je naše hledaná inverzní matice A^{-1}.

Ukázka výpočtu inverzní matice:

$$\left[\begin{array}{cc|cc} 1 & 3 & 1 & 0 \\ 2 & 4 & 0 & 1 \end{array}\right] \begin{array}{c} \\ -2r_1 \end{array} \mapsto \left[\begin{array}{cc|cc} 1 & 3 & 1 & 0 \\ 0 & -2 & -2 & 1 \end{array}\right] \begin{array}{c} \frac{3}{2}r_2 \\ \\ \end{array} \mapsto \left[\begin{array}{cc|cc} 1 & 0 & -2 & \frac{3}{2} \\ 0 & -2 & -2 & 1 \end{array}\right] \begin{array}{c} \\ \frac{1}{2}r_2 \end{array} \mapsto$$
$$\left[\begin{array}{cc|cc} 1 & 0 & -2 & \frac{3}{2} \\ 0 & 1 & 1 & -\frac{1}{2} \end{array}\right]$$

K matici A = $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$ *tedy existuje inverzní matice* $A^{-1} = \begin{bmatrix} -2 & \frac{3}{2} \\ 1 & -\frac{1}{2} \end{bmatrix}$.

Podrobnější návod, jak získat inverzní matici naleznete například v [1]. Algoritmus, který je zde použit, si vysvětlíme v kapitole 4.

3.4 Podmíněnost matic

Pro určení čísla podmíněnosti $\kappa(A)$ matice A potřebujeme znát pojmy vektorový prostor, norma vektoru a norma matice.

3.4.1 Vektorový prostor

Definice 3.4.1. Vektorovým prostorem nad množinou reálných čísel $\mathbb{R}$ je neprázdná množina $\mathcal{V}$, která má definovány dvě operace.

- Sčítání vektorů: Každé dvojici vektorů $a, b \in \mathcal{V}$ přiřadí vektor $a + b \in \mathcal{V}$.
- Násobení vektoru číslem: Každému číslu $k \in \mathbb{R}$ a vektoru $a \in \mathcal{V}$ přiřadí vektor $ka \in \mathcal{V}$.

Tyto operace musí splňovat podmínky, které naleznete např. v [1].

3.4.2 Norma vektoru

Definice 3.4.2. Norma vektoru $\|\cdot\|$ je funkce na vektorovém prostoru $\mathcal{V}$, která každému nenulovému vektoru přiřadí reálné kladné číslo. Norma vektoru musí pro dva libovolné vektory $x, y \in \mathcal{V}$ a libovolné číslo k splňovat:

- $\|x\| > 0$, $\|x\| = 0 \Leftrightarrow x = 0$
- $\|x + y\| \leq \|x\| + \|y\|$
- $\|kx\| = |k| \cdot \|x\|$

Nejznámějším příkladem normy je tzv. **Euklidovská norma** definována následovně:

$$\|(x_1, x_2, \cdots, x_n)\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

3.4.3 Norma matice

Definice 3.4.3. Norma matice $\|\cdot\|$ je funkce na prostoru všech matic, která každé nenulové matici přiřadí reálné kladné číslo. Norma matice musí pro dvě libovolné reálné matice A,B a libovolné číslo α splňovat:

- $\|A\| > 0$, $\|A\| = 0 \Leftrightarrow A$ je nulová matice
- $\|\alpha A\| = |\alpha|\|A\|$
- $\|A + B\| \leq \|A\| + \|B\|$

Poznámka. Pro některé maticové normy lze odvodit:

$$\|AB\| \leq \|A\| \cdot \|B\| \tag{3.1}$$

Frobeniova (někdy také Euklidovská) norma matice je definována následovně:

$$\|A\| = \sqrt{\sum_i \sum_j a_{ij}^2}$$

3.4.4 Podmíněnost matice

Při řešení soustav rovnic se setkáváme s výsledky, které jsou velmi přesné i přes vliv zaokrouhlovacích chyb, ale i s výsledky, kde mohou mít zaokrouhlovací chyby velký vliv. Lze soudit, že to bude způsobeno zvolenými koeficienty matice. Podrobnější popis o stabilitě výpočetních úloh lze nalézt v [3].

Definice 3.4.4. Číslo podmíněnosti matice A, kde $A \in \mathbb{R}^{n \times n}$ je čtvercová regulární matice, spočítáme jako:

$$\kappa_p(A) = \|A\|\|A^{-1}\|,$$

kde $\|\cdot\|$ značí libovolnou maticovou normu (nejčastěji Euklidovu).

Věta 3.4.5. Mějme matici A, která je regulární čtvercová matice řádu n, a vektory b a x jsou nenulové n-složkové vektory. Nechť platí:

$$Ax = b$$

Dále mějme vektory $\widetilde{b}$ a $\widetilde{x}$, které jsou také n-složkové vektory a nechť pro ně platí:

$$A\widetilde{x} = \widetilde{b}$$

Potom

$$\frac{\|x - \widetilde{x}\|}{\|x\|} \leq \kappa(A) \frac{\|b - \widetilde{b}\|}{\|b\|} \tag{3.2}$$

Důkaz. Nechť platí

$$b = Ax \ \ resp. \ \ x - \widetilde{x} = A^{-1}(b - \widetilde{b}) \tag{3.3}$$

pomocí (3.1) dostaneme:

$$\|x\|^{-1} \leq \|A\|\|b\|^{-1} \ \ resp. \ \ \|x - \widetilde{x}\| \leq \|A^{-1}\|^{\|}b - \widetilde{b}\| \tag{3.4}$$

Z těchto tvrzení získáme (3.2).

Číslo podmíněnosti nebo také číslo podmíněnosti matice, je číslo, které do značné míry předpovídá chování (zvláště přesnost) řady numerických maticových algoritmů.

Dobře podmíněnou soustavou lineárních rovnic, budeme nazývat takovou soustavu, kde prvky v hlavní diagonále matice A této soustavy dominují, tedy jsou v absolutní hodnotě větší než ostatní prvky v příslušném řádku nebo sloupci. Také lze dokázat, že stejný výsledek dostaneme v případě, má-li matice A a k ní inverzní matice A^{-1} přibližně stejné v absolutní hodnotě největší prvky. Samozřejmě determinant dobře podmíněné matice musí být nenulový!

Špatně podmíněná soustava rovnic má rovněž determinant nenulový, ale liší se v absolutní hodnotě největší prvky matic A a A^{-1} až o řády. Řešení této soustavy, může vést k vytvoření rezidua.

Reziduum získáme z předpisu:

$$r = A\widetilde{x} - b \tag{3.5}$$

Pokud je $r = 0$, pak máme přesné řešení.

Číslo podmíněnosti matice, lze získat v MatLabu příkazem $cond(A)$.

3.5 Klasifikace matice

Klasifikování matic se provádí pouze pro čtvercové symetrické reálné matice. Abychom mohli matici klasifikovat, je nutné ji upravit do diagonálního tvaru.
Tuto úpravu provedeme pomocí tzv. **elementárních kongruencí**. Více o úpravách v [1].

Věta 3.5.1. Pozitivně definitní matice je taková matice A, která v upraveném diagonálním tvaru má na diagonále pouze **nenulová kladná** čísla.

Další výsledky klasifikace matic, se kterými v této práci nesetkáme, jsou negativně definitní, pozitivně semidefinitní, negativně semidefinitní a indefinitní matice. Více si o klasifikaci matic můžete přečíst např. v [1].

3.6 Matice se známou inverzí

Pro některé regulární matice lze najít explicitní předpis pro spočítání jejich inverze. Takové matice se nazývají matice se známou inverzí.
Mezi matice se známou inverzí patří například Pascalova, Hilbertova nebo Vandermondova matice.

3.6.1 Pascalova matice

Pascalova matice je špatně podmíněná, čtvercová, regulární, pozitivně definitní matice. Pascalova symetrická matice vznikne součinem dolní trojúhelníkové matice P_n^L a horní troúhelníkové matice P_n^U, které představují pascalův trojúhelník.

Pascalovu matici získáme v MatLabu příkazem $pascal(n)$, kde n je rozměr čtvercové matice.

Ukázka vzniknutí Pascalovy matice o rozměru $n = 5$:

$$P_5^L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}, P_5^U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 3 & 6 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_5 = P_5^L P_5^U = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{bmatrix}$$

3.6.2 Hilbertova matice

Tato matice je velmi špatně podmíněná, čtvercová, regulární, pozitivně definitní, symetrická matice, jejíž první řádek je tvořen začátkem harmonické řady, tzn. převrácenými hodnotami přirozených čísel. Další řádky matice vznikají posunem v harmonické řadě o jednu pozici vpravo.

Hilbertovu matici získáme v MatLabu příkazem $hilb(n)$, kde n je rozměr čtvercové matice.

Pro představu si uvedeme matice H_3 a H_5.

$$H_3 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}, H_5 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}$$

3.6.3 Vandermondova matice

Vandermondova matice je špatně podmíněná, čtvercová, regulární, pozitivně definitní matice. Vandermondovu matici získáme z řádkového vektoru $x = [x_1, x_2, \cdots, x_n]$. Pro zajištění pozitivně definitní matice, budeme matice generovat z vektoru $[1, 2, 3, ..., n]$.

Matice se vytvoří podle následujícího algoritmu:

$$V_n = \begin{bmatrix} x_1^0 & x_1^1 & \cdots & x_1^{n-2} & x_1^{n-1} \\ x_2^0 & x_2^1 & \cdots & x_2^{n-2} & x_2^{n-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_n^0 & x_n^1 & \cdots & x_n^{n-2} & x_n^{n-1} \end{bmatrix}$$

Pro představu si uvedeme matice V_3 a V_5.

$$V_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix}, V_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \\ 1 & 5 & 25 & 125 & 625 \end{bmatrix}$$

Kapitola 4

Gaussova eliminační metoda

GEM je jedna z nejznámějších a nejpoužívanějších metod při řešení soustav lineárních rovnic. Tato metoda vede k (teoreticky) přesnému řešení v konečném počtu kroků.
GEM se používá také pro výpočet inverzní matice nebo výpočet determinantu. Tato kapitola je podpořena studijní literaturou [1].

4.1 Postup výpočtu

4.1.1 Úprava na schodový tvar

Pomocí elementárních řádkových operací můžeme převést matici (2.2) na tzv. **schodový tvar**, tj. na tvar, v němž jsou jako první nenulové prvky řádků zvané **pivoty** uspořádány jako schody klesající zleva doprava. Důležitý požadavek je, aby pivoty nebyly nad sebou a aby všechny případné nulové řádky byly umístěny dole.

Příklady schodových matic:

$$\mathrm{A} = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}, \mathrm{B} = \begin{bmatrix} 0 & 2 & 2 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}, \mathrm{C} = \begin{bmatrix} 0 & 3 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathrm{D} = \begin{bmatrix} 4 & 3 & 2 \\ 0 & 2 & 1 \\ 0 & 0 & 5 \end{bmatrix}$$

Při úpravě matice využijeme pozorování, že je-li v matici (2.2) prvek a_{ij} nenulový, pak vynásobíme-li i-tý řádek této matice číslem $-a_{kj}/a_{ij}$ a přičteme-li ho ke k-tému řádku, bude mít upravená matice v k-tém řádku a j-tém sloupci prvek:

$$a_{kj} + (-a_{kj}/a_{ij})a_{ij} = 0 \tag{4.1}$$

Pokud je prvek a_{11} nenulový, lze takto upravit matici (2.1) na tvar

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{m2}^{(1)} & \cdots & a_{mn}^{(1)} & b_m^{(1)} \end{array}\right] \tag{4.2}$$

Pokud bude také prvek $a_{22}^{(1)}$ nenulový, můžeme obdobně dosáhnout pomocí elementárních řádkových operací, aby i pod ním byly v upravené matici nuly. Bude-li pokaždé $a_{ii}^{(i-1)} \neq 0$, dostaneme nakonec matici (4.3) ve schodovém tvaru s nenulovými prvky a_{11},$a_{22}^{(1)}$,$\cdots$,$a_{kk}^{(k-1)}$.

$$\left[\begin{array}{cccccc|c} a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2k}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} & b_k^{(k-1)} \\ 0 & 0 & \cdots & 0 & \cdots & 0 & b_{k+1}^{(k)} \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \end{array}\right] \tag{4.3}$$

Můžeme si všimnout, že rozložení nenulových prvků v levé části upravené matice soustavy připomíná trojúhelník, proto říkáme, že **matice je v trojúhelníkovém tvaru**. Úpravu na trojúhelníkový tvar lze provést, i v případě, kdy $a_{ii}^{(i-1)} = 0$, pokud je možno najít prvek $a_{ji}^{(i-1)} \neq 0$, $j > i$. Stačí vzájemně vyměnit před úpravou i-tý a j-tý řádek. Tato úprava se nazývá pivotizace a více si o ní řekneme v kapitole (4.3).

Každou matici však nelze elementárními řádkovými úpravami převést na trojúhelníkový tvar. Kdyby byl například první sloupec celý nulový, nebyli bychom žádnou řádkovou úpravou schopni zajistit, aby se do levého horního rohu dostal nenulový prvek. V takových případech přeskočíme nulový sloupec a začneme pracovat s prvním nenulovým sloupcem. Podobně bychom pokračovali s úpravou dalších řádků. Nedospěli bychom však k matici ve tvaru (4.3), ale k obecnější matici ve schodovém tvaru.

4.1.2 Zpětná substituce

Teď si ukážeme, jak získat řešení soustavy lineární rovnice s maticí ve schodovém tvaru. Můžou nastat tři případy, které jsme zmínili již v kapitole (2.1). Vzhledem k tomu, že se budeme zabývat soustavou lineárních rovnic s regulární maticí, tak nás zajímá pouze jeden možný výsledek:

Definice 4.1.1. Rozšířená matice má trojúhelníkový tvar (4.3) s $k = n$, $b_{n+1}^{(n)} = 0$ a $a_{ii}^{(i-1)} \neq 0$, $i = 1, \cdots, n$. Pak n-tá rovnice má tvar

$$a_{nn}^{(n-1)} x_n = b_n^{(n-1)} \tag{4.4}$$

ze kterého snadno vypočteme x_n. Po dosazení do předchozích rovnic zbude v $(n-1)$-ní rovnici jediná neznámá, kterou také snadno spočteme. Budeme-li takto dále postupovat určíme **řešení soustavy**.

Příklad soustavy s jediným řešením:

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 10 \\ 2 & 3 & 1 & 13 \\ 3 & 1 & 2 & 13 \end{array}\right] \begin{array}{c} \\ -2r_1 \\ -3r_1 \end{array} \mapsto \left[\begin{array}{ccc|c} 1 & 2 & 3 & 10 \\ 0 & -1 & -5 & -7 \\ 0 & -5 & -7 & -17 \end{array}\right] \begin{array}{c} \\ \mapsto \\ -5r_2 \end{array} \left[\begin{array}{ccc|c} 1 & 2 & 3 & 10 \\ 0 & -1 & -5 & -7 \\ 0 & 0 & 18 & 18 \end{array}\right] \quad (4.5)$$

Není těžké z této soustavy dostat její řešení pomocí zpětné substituce. Z posledního řádku dostaneme rovnici $18x_3 = 18$, tedy $x_3 = 1$. Dosazením do předchozího řádku získáme rovnici $-x_2 - 5x_3 = -7$, tedy $x_2 = 2$ a nakonec z prvního řádku získáme rovnici $x_1 + 2x_2 + 3x_3 = 10$, z čehož po dosazení a úpravě zjistíme, že $x_1 = 3$. Tato soustava má skutečně jediné řešení, a to vektor $x = [3, 2, 1]^T$

4.2 GEM bez pivotizace

GEM bez pivotizace je řešení soustavy lineárních rovnic bez záměny pořadí řádků. Předpokládáme tedy, že při úpravách nikdy nenastane případ $a_{ii}^{(i-1)} = 0$. Postupným aplikováním ekvivalentních úprav získáváme schodovou matici a z ní pomocí zpětné substituce výsledný vektor. Nyní si zapíšeme algoritmus.

4.2.1 Algoritmus

Algoritmus GEM je rozdělen na dvě části - dopředný a zpětný chod. Dopředný chod je úprava, kdy získáváme ze zadané matice *A* horní trojúhelníkovou matici *U*, tedy soustavu $Ax = b$ upravíme na soustavu $Ux = y$. Zpětný chod je výpočet řešení původní soustavy ze soustavy $Ux = y$.

Dopředný chod GEM

Popis algoritmu:
Vstupními parametry jsou matice A a vektor pravé strany b. Výstupem je matice U a vektor pravé strany y.

Algoritmus bude probíhat následovně:

Vnější cyklus bude provádět tzv. fáze. V k-té fázi, $1 \leq k \leq n-1$, se provádí eliminace v k-tém sloupci matice. Prvky matice na začátku k-té fáze označíme jako $a_{ij}^{(k)}$ a prvky vektoru pravé strany jako $a_{i(n+1)}^{(k)}$, kde i,j určují aktuální pozici. Tedy na začátku 1.fáze je $a_{ij}^{(1)} = a_{ij}$ a $a_{i(n+1)}^{(1)} = b_i$. Eliminace v k-tém sloupci probíhá pod jejím diagonálním prvkem $a_{kk}^{(k)}$, který se nazývá pivot k-té fáze. Je nutné, abychom si prvně spočítali multiplikátory k-té fáze, které získáme z předpisu:

$$m_{ik} = -\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, i = k+1, ..., n \quad (4.6)$$

Následně přičteme m_{ik}-násobek k-tého řádku k řádku i-tému, tedy:

$$a_{ij}^{(k+1)} := a_{ij}^{(k)} + m_{ij}a_{kj}^{(k)}, j = k+1, ..., n+1 \tag{4.7}$$

pro $i = k+1, ..., n+1$.

Algoritmus:

Vstup: $A = (a_{ij}), \quad b = (b_i)$.

$a_{ij}^{(1)} = a_{ij}, \quad a_{in+1}^{(1)} = b_i, \quad i, j = 1, .., n.$

Pro $k = 1, ..., n-1$ proveď k-tou fázi:

Pro $i = k+1, ..., n$ přičti m_{ik}-násobek k-tého řádku k i-tému řádku:
$m_{ik} := -a_{ik}^{(k)}/a_{kk}^{(k)}$
Pro $j = k+1, ..., n+1$ proveď přičítání v j-tém sloupci:
$a_{ij}^{k+1} := a_{ij}^{k} + m_{ik}a_{ij}^{k}$

Polož $u_{ij} := a_{ij}^{(n)}$ pro $i \leq j, \quad u_{ij} := 0$ pro $i > j, \quad y_i = a_{in+1}^{(n)}, \quad i, j = 1, ..., n.$

Výstup: $U = (u_{ij}), \quad y = (y_i)$

Zpětný chod GEM

Popis algoritmu:
Vstupními parametry jsou matice U a vektor pravé strany y. Výstupem je výsledný vektor x, který je řešením dané soustavy.

Pracujeme se soustavou lineárních rovnic $Ux = y$, kde U je horní trojúhelníková matice, $U = (u_{ij}), u_{ij} = 0, i > j$, a vektorem pravé strany y = (y_i).
Soustava má nasledující tvar:

$$\begin{aligned} u_{11}x_1 + u_{12}x_2 + ... + u_{1n}x_n &= y_1, \\ u_{22}x_2 + ... + u_{2n}x_n &= y_2, \\ &\cdots \\ u_{nn}x_n &= y_n. \end{aligned}$$

Výpočet začínáme od poslední rovnice, kde je pouze jedna neznámá. Po získání proměnné x_n ji dosadíme do předchozí rovnice a spočítáme neznámou x_{n-1}. Takto budeme postupovat až získáme neznámou x_1. Celkem vyřešíme n rovnic a dostaneme vektor $x = [x_1, x_2, ..., x_n]^T$.

Algoritmus:

Vstup: $U = (u_{ij}),\ \ y = (y_i)$.

$x_n := y_n/u_{nn}$.

Pro $i = n-1, ..., 1$ počítej s i-tou neznámou:

$$x_i := (y_i - u_{in}x_n - ... - u_{i(i+1)}x_{i+1})/u_{ii}.$$

Výstup: $x = (x_i)$

4.3 GEM s pivotizací

4.3.1 Pivotizace u GEM

Výběr pivota můžeme provést dvěma algoritmy. Buď provedeme **částečnou** pivotizaci nebo **úplnou** pivotizaci. U GEM budeme používat pouze částečnou pivotizaci.

Částečná pivotizace

V každém aktuálně eliminovaném sloupci budeme postupovat následujícím způsobem:

- Najdeme v absolutní hodnotě maximální prvek a_{ij}, kde $i < j \leq n$.
- Zaměníme řádek s nalezeným maximální prvkem s řádkem, kde požadujeme pivot.

Průběh této pivotizace je znázorněn na následujícím příkladě:

$$\left[\begin{array}{ccc|c} 2 & 2 & 3 & 10 \\ 1 & 3 & 1 & 13 \\ 3 & 1 & 2 & 13 \end{array}\right] \begin{array}{c} r_3 \\ \\ r_1 \end{array} \mapsto \left[\begin{array}{ccc|c} 3 & 1 & 2 & 13 \\ 1 & 3 & 1 & 13 \\ 2 & 2 & 3 & 10 \end{array}\right]$$

Pivotizace proběhla mezi řádky r_1 a r_3. Tímto zabráníme možnosti, kdy je pivot a_{ii} nulový.
Pro představu si ukážeme příklad řešení soustavy lineárních rovnic pomocí GEM s pivotizací.

Příklad soustavy s pivotizací

$$\left[\begin{array}{ccc|c} 2 & 1 & -3 & 0 \\ 6 & 3 & -8 & 0 \\ 2 & -1 & 5 & -4 \end{array}\right] \begin{array}{c} r_2 \\ r_1 \\ \\ \end{array} \mapsto \left[\begin{array}{ccc|c} 6 & 3 & -8 & 0 \\ 2 & 1 & -3 & 0 \\ 2 & -1 & 5 & -4 \end{array}\right] \begin{array}{c} \\ -\frac{2}{6}r_1 \\ -\frac{2}{6}r_1 \end{array} \mapsto$$
$$\left[\begin{array}{ccc|c} 6 & 3 & -8 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -2 & \frac{23}{3} & -4 \end{array}\right] \begin{array}{c} \\ r_3 \\ r_2 \end{array} \mapsto \left[\begin{array}{ccc|c} 6 & 3 & -8 & 0 \\ 0 & -2 & \frac{23}{3} & -4 \\ 0 & 0 & -\frac{1}{3} & 0 \end{array}\right] \tag{4.8}$$

Z matice ve schodovém tvaru získáme výsledný vektor pomocí zpětného chodu z kapitoly (4.2.1). Výsledkem této soustavy je tedy vektor $x = [-1, 2, 0]^T$.

4.3.2 Algoritmus GEM s pivotizací

Algoritmus GEM s pivotizací se liší od algoritmu GEM bez pivotizace pouze ve výběru pivota, který se provádí v dopředném chodu GEM 4.2.1 tak, že na začátek k-té fáze vsuneme tyto řádky:

- Najdi p, $p \geq k$, takové, že $|a_{pk}^{(k)}| = max\{|a_{ik}^{(k)}|, i \geq k\}$;
- Zaměň p-tý a k-tý řádek matice v k-té fázi.

Algoritmus:

Vstup: $A = (a_{ij}),\ \ b = (b_i)$.

$a_{ij}^{(1)} = a_{ij},\ \ a_{in+1}^{(1)} = b_i,\ \ i, j = 1, .., n$

Pro $k = 1, ..., n-1$ proveď k-tou fázi:

Najdi p, $p \geq k$, takové, že $|a_{pk}^{(k)}| = max\{|a_{ik}^{(k)}|, i \geq k\}$

Prohoď p-tý a k-tý řádek matice v k-té fázi

Pro $i = k+1, ..., n$ přičti m_{ik}-násobek k-tého řádku k i-tému řádku:

$m_{ik} := -a_{ik}^{(k)}/a_{kk}^{(k)}$

Pro $j = k+1, ..., n+1$ proveď přičítání v j-tém sloupci:

$a_{ij}^{k+1} := a_{ij}^{k} + m_{ik}a_{ij}^{k}$

Polož $u_{ij} := a_{ij}^{(n)}$ pro $i \leq j$, $\ u_{ij} := 0$ pro $i > j$, $\ y_i = a_{in+1}^{(n)}$, $\ i, j = 1, ..., n$.

Výstup: $U = (u_{ij}),\ \ y = (y_i)$

Kapitola 5

Nevillova eliminační metoda

Obdobně jako u GEM získáme pomocí elementárních řádkových operací matici ve schodovém tvaru (4.3). Je zde ovšem jedna zásadní změna. U GEM jsme eliminovali pomocí **pivota**, který byl pevně vybrán pro celý sloupec. U NEM se nám bude pivot měnit v každém řádku. Kapitola je podpořena studijním materiálem [5].

5.1 Postup výpočtu

5.1.1 Úprava na schodový tvar

V matici (2.2) začneme eliminaci od pozice a_{n1}, kterou budeme eliminovat pomocí prvku, který se nachází nad ní, tedy pomocí prvku $a_{(n-1)1}$. Pak eliminujeme prvek $a_{(n-1)1}$ pomocí prvku $a_{(n-2)1}$. Takto pokračujeme až provedeme eliminaci prvku a_{21} pomocí prvku a_{11}.

Pokud budou všechny prvky v eliminovaném sloupci nenulové, tak upravená matice bude vypadat následovně:

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{m2}^{(1)} & \cdots & a_{mn}^{(1)} & b_m^{(1)} \end{array}\right]$$

Budou-li po každé další úpravě prvky v eliminovaném sloupci nenulové, získáme tuto matici:

$$\left[\begin{array}{cccccc|c} a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \cdots & a_{2k}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} & b_k^{(k-1)} \\ 0 & 0 & \cdots & 0 & \cdots & 0 & b_{k+1}^{(k)} \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \end{array}\right]$$

Vidíme, že jsme získali schodovou matici (4.3) s nenulovými prvky a_{11},$a_{22}^{(1)}$,$\cdots$,$a_{kk}^{(k-1)}$. V případě, že narazíme v průběhu eliminace na nulový pivot je nutné provést pivotizaci (5.3.1).

5.1.2 Zpětná substituce

Zpětná substituce u NEM probíhá stejně jako u GEM. Její popis tedy naleznete v kapitole (4.1.2). Lepší představu o NEM získáme z následujícího jednoduchého příkladu.

Řešení soustavy rovnic pomocí Nevillovy eliminace

Vezmeme si příklad (4.5).

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 10 \\ 2 & 3 & 1 & 13 \\ 3 & 1 & 2 & 13 \end{array}\right] \begin{array}{c} \\ \\ -\frac{3}{2}r_2 \end{array} \mapsto \left[\begin{array}{ccc|c} 1 & 2 & 3 & 10 \\ 2 & 3 & 1 & 13 \\ 0 & -\frac{7}{2} & \frac{1}{2} & -\frac{13}{2} \end{array}\right] \begin{array}{c} \\ -2r_1 \\ \\ \end{array} \mapsto$$

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 10 \\ 0 & -1 & -5 & -7 \\ 0 & -\frac{7}{2} & \frac{1}{2} & -\frac{13}{2} \end{array}\right] \begin{array}{c} \\ \\ -\frac{7}{2}r_2 \end{array} \mapsto \left[\begin{array}{ccc|c} 1 & 2 & 3 & 10 \\ 0 & -1 & -5 & -7 \\ 0 & 0 & 18 & 18 \end{array}\right] \tag{5.1}$$

Když se podíváme na příklad (4.5) a (5.1) tak vidíme, že výsledná matice je stejná. Je zřejmé, že Gaussovou eliminační metodou i Nevillovou eliminační metodou dojdeme ke stejnému řešení, k vektoru $x = [3, 2, 1]^T$.

5.2 NEM bez pivotizace

Při úpravě předpokládáme, že nenarazíme na případ, že by pivot $a_{ij}^{(j)}$, $j = 1, ..., n-1, i = j, ..., n$ byl nulový. Postupným aplikováním ekvivalentních úprav získáme schodovou matici a z ní pomocí zpětné substituce výsledný vektor, který je řešením soustavy. Postup jsme si předvedli na příkladě (5.1), nyní si popíšeme algoritmus.

5.2.1 Algoritmus

Algoritmus NEM má stejně jako GEM dvě části - dopředný a zpětný chod.

Dopředný chod NEM

Popis algoritmu:
Vstupními parametry jsou matice A a vektor pravé strany b. Výstupem je matice U a vektor pravé strany y.

Algoritmus bude probíhat následovně:

Algoritmus začíná výběrem k-tého sloupce, kde $k = 1, ..., n-1$. Poté vybereme i-tý řádek, $i = n, ..., k+1$. Pro každý i-tý řádek si spočítáme multiplikátor pomocí předpisu:

$$m_{ik} := -a_{ik}^{(k)}/a_{(i-1)k}^{(k)} \tag{5.2}$$

Poté pro j-tý sloupec, kde $j = k, ..., n+1$ provedeme přičtení m_{ik}-násobku řádku s pivotem $a_{(i-1)j}$ k řádku s pivotem a_{ij}, tedy:

$$a_{ij}^{(k+1)} := a_{ij}^{(k)} + m_{ik}a_{(i-1)j}^{(k)} \tag{5.3}$$

Algoritmus:

Vstup: $A = (a_{ij}),\ \ b = (b_i)$.

$a_{ij}^{(1)} = a_{ij},\ \ a_{in+1}^{(1)} = b_i,\ \ i, j = 1, .., n$

Pro $k = 1, ..., n-1$ proveď:

Pro $i = n, ..., k+1$ přičti m_{ik}-násobek $(i-1)$-ního řádku k i-tému řádku:

$m_{ik} := -a_{ik}^{(k)}/a_{(i-1)k}^{(k)}$

Pro $j = k, ..., n+1$ proveď přičítání v j-tém sloupci:

$a_{ij}^{(k+1)} := a_{ij}^{(k)} + m_{ik}a_{(i-1)j}^{(k)}$

Polož $u_{ij} := a_{ij}^{(n)}$ pro $i \leq j$, $\ u_{ij} := 0$ pro $i > j$, $\ y_i = a_{in+1}^{(n)},\ \ i, j = 1, ..., n$.

Výstup: $U = (u_{ij}),\ \ y = (y_i)$.

Zpětný chod NEM

Postup je totožný s výpočtem zpětného chodu GEM.

5.3 NEM s pivotizací

5.3.1 Pivotizace u NEM

Již dříve jsme zmínili dva způsoby pivotizace - částečnou a úplnou pivotizaci. U NEM budeme používat úplnou pivotizaci.

Úplná pivotizace

- V části sloupce, kde provádíme eliminaci v k-té fázi, najdeme nerostoucí uspořádání prvků podle jejich absolutních hodnot.
- Zaměníme pořadí řádků podle nalezeného uspořádání.

Před popisem algoritmus NEM s pivotizací si tento postup ukážeme na příkladě (4.8).

Příklad NEM s pivotizací

$$
\left[\begin{array}{ccc|c} 2 & 1 & -3 & 0 \\ 6 & 3 & -8 & 0 \\ 2 & -1 & 5 & -4 \end{array}\right] \begin{array}{c} r_2 \\ r_1 \\ \end{array} \mapsto \left[\begin{array}{ccc|c} 6 & 3 & -8 & 0 \\ 2 & 1 & -3 & 0 \\ 2 & -1 & 5 & -4 \end{array}\right] \begin{array}{c} \\ \mapsto \\ -r_2 \end{array} \left[\begin{array}{ccc|c} 6 & 3 & -8 & 0 \\ 2 & 1 & -3 & 0 \\ 0 & -2 & 8 & -4 \end{array}\right] -\tfrac{1}{6}r_1
$$

$$
\mapsto \left[\begin{array}{ccc|c} 6 & 3 & -8 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -2 & \frac{23}{3} & -4 \end{array}\right] \begin{array}{c} \\ r_3 \\ r_2 \end{array} \mapsto \left[\begin{array}{ccc|c} 6 & 3 & -8 & 0 \\ 0 & -2 & \frac{23}{3} & -4 \\ 0 & 0 & -\frac{1}{3} & 0 \end{array}\right] \tag{5.4}
$$

Výsledek této soustavy je vektor $x = [-1, 2, 0]^T$. Při porovnání výsledků z příkladů (4.8) a (5.4) vidíme, že jsme došli ke stejnému řešení jako při použití GEM s pivotizací.

5.3.2 Algoritmus

Dopředný chod NEM

Výpočet dopředného chodu NEM s pivotizací se liší od NEM bez pivotizace pouze ve způsobu výběru pivota, který se provádí v dopředném chodu NEM 5.2.1 tak, že se na začátek k-té fáze vsunou tyto řádky:

- Najdi pořadí prvků $|a_{ik}^{(k)}|$, $i = k, ...n$ takové, že budou seřazeny od největšího po nejmenší.
- Prohoď pořadí j-tého až n-tého řádku podle nalezeného pořadí.

Algoritmus:

Vstup: $A = (a_{ij}),\ \ b = (b_i)$.

$a_{ij}^{(1)} = a_{ij},\ \ a_{in+1}^{(1)} = b_i,\ \ i, j = 1, .., n$

Pro $k = 1, ..., n-1$ proveď:

Najdi pořadí prvků $|a_{ik}^{(k)}|$, $i = k, ...n$ takové, že budou seřazeny vzestupně.

Prohoď pořadí k-tého až n-tého řádku podle nalezeného pořadí.

Pro $i = n, ..., k+1$ přičti m_{ik}-násobek $(i-1)$-ního řádku k i-tému řádku:

$m_{ik} := -a_{ik}^{(k)}/a_{(i-1)k}^{(k)}$

Pro $j = k, ..., n+1$ proveď přičítání v j-tém sloupci:

$a_{ij}^{(k+1)} := a_{ij}^{(k)} + m_{ik} a_{(i-1)j}^{(k)}$

Polož $u_{ij} := a_{ij}^{(n)}$ pro $i \le j$, $u_{ij} :=$ 0 pro $i > j$, $y_i = a_{in+1}^{(n)}$, $i, j = 1, ..., n$.

Výstup: $U = (u_{ij}),\ \ y = (y_i)$.

Kapitola 6

Numerické experimenty

Numerické experimenty s GEM a NEM budeme provádět na úlohách s totálně pozitivní maticí.

Definice 6.0.1. Totálně pozitivní maticí budeme nazývat matici řádu $n \times n$, pro kterou platí:

$$\det A_i > 0,$$

kde A_i je libovolná čtvercová submatice matice A.

Jinými slovy, totálně pozitivní matice má kladné minory (více o minorech v [1]). Mezi totálně pozitivní matice patří dříve zmíněné matice se známou inverzí - Pascalova, Vandermondova a Hilbertova.

6.1 Nepřesnosti při počítání

6.1.1 Číslo s plovoucí řádovou čárkou

Při použití počítače pro výpočty s požadavkem na velkou přesnost se setkáváme s následujícím problémem. Počítač není schopen zapsat nekonečný počet cifer a je omezen datovými typy. Největší číslo, které je tedy schopen vyjádřit může mít až 16 cifer. Proto se velmi často využívají čísla s plovoucí řádovou čárkou. Čísla s plovoucí řádovou čárkou zapisujeme jako:

$$m \times z^e, \tag{6.1}$$

kde

- m... je mantisa
- z... je základ soustavy (nejčastěji 2)
- e... je exponent

6.1.2 Přesnost výpočtu na počítači

Přesnost počítače je dána tzv. strojovým epsilon ϵ. Pro čísla s plovoucí čárkou se využívají dva typy čísel:

Real float ... neboli jednoduchá přesnost, $\epsilon = 2^{-23}$, což je v desítkové soustavě přibližně rovno hodnotě 1.19×10^{-7}. Toto číslo je reprezentováno zhruba 6 desetinnými místy.

Double float ... neboli dvojitá přesnost, $\epsilon = 2^{-52}$, což je v desítkové soustavě přibližně rovno hodnotě 2.22×10^{-16}. Toto číslo je reprezentováno zhruba 15 desetinnými místy.

6.2 Chyby

K chybám vzniklým z numerických výpočtů dochází aproximací čísla, kde při realizaci výpočtu nahrazujeme reálnou (přesnou) hodnotu čísla x jeho přibližnou hodnotou $\tilde{x}$. Tímto nahrazením se dopouštíme zaokrouhlovacích chyb, které mohou výrazně ovlivnit přesnost výsledku.

6.2.1 Zaokrouhlovací chyby

Definice 6.2.1. Mějme přesnou hodnotu x a její aproximaci $\tilde{x}$.
Rozdíl $\triangle x = x - \tilde{x}$ nazýváme *absolutní chybou* aproximace $\tilde{x}$. Hodnotu $\epsilon(x) \geq 0$ takovou, že $|x - \tilde{x}| = |\triangle x| \leq \epsilon(\tilde{x})$ nazýváme *odhadem absolutní chyby*.

Číslo $\frac{\triangle x}{x} = \frac{x-\tilde{x}}{x}, x \neq 0$ nazýváme *relativní chybou* aproximace $\tilde{x}$. Hodnotu $\delta(x) \geq 0$ takové, že $|\frac{x-\tilde{x}}{x}| \leq \frac{\epsilon(\tilde{x})}{|x|} = \delta(\tilde{x})$ nazýváme *odhad relativní chyby*.

V této práci budeme určovat numerické chyby my_eps_1 a my_eps_2. Pro určení chyby my_eps_1 budeme potřebovat řešení soustavy získané pomocí zpětného lomítka v MatLabu, tedy $y = A \setminus b$ a řešení soustavy pomocí mnou implementovaného algoritmu, kde řešením je vektor x. Chybu poté spočítáme jako (6.2).

$$my\_eps_1 = \frac{||y - x||}{||y||}, \tag{6.2}$$

V druhém případě se obejdeme bez vestavěných funkcí MatLabu. K určení chyby my_eps_2 budeme potřebovat reziduum r, které získáme ze vzorce (3.5) a vektor původní pravé strany b. Poté chybu spočítáme jako (6.3).

$$my\_eps_2 = \frac{||r||}{||b||}, \tag{6.3}$$

6.3 Testy

Testování přesnosti výpočtu algoritmů GEM a NEM provádíme na soustavě lineárních rovnic s totálně pozitivní maticí, konkrétně na maticích se známou inverzí, které jsme uvedli v kapitole 3.6. Testy jsou zaměřeny na porovnání výpočtů mnou implementovaných algoritmů GEM, GEM s pivotizací, NEM a NEM s pivotizací. Podle [5] budeme očekávat, že nejpřesnějších výsledků dosáhneme pomocí algoritmu NEM s pivotizací, o něco méně přesný by měl být GEM s pivotizací, následovaný NEM a nejmenší přesnost očekáváme u GEM. Očekávání největší přesnosti pro NEM s pivotizací je zapříčiněno tím, že při výpočtu multiplikátoru dělíme velikostně srovnatelná čísla. Tím bychom měli docílit menších numerických chyb.

Výsledky testů pro soustavu lineárních rovnic s určitou maticí se známou inverzí jsou vždy rozděleny do dvou tabulek. Tabulky obsahují tyto sloupce:

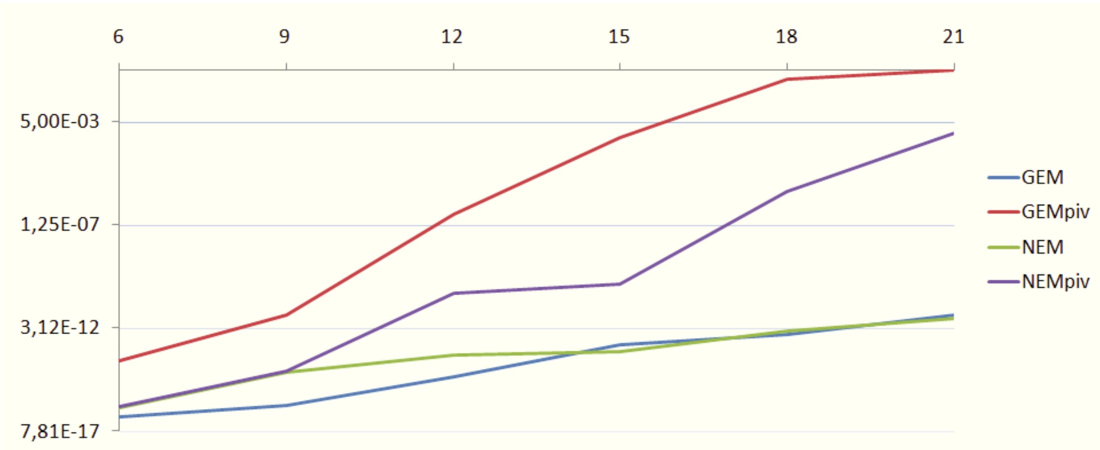
n	- rozměr matice A
cond(A)	- číslo podmíněnosti matice A vypočítané MatLabem
GEM	- přesnost výsledků dosažených pomocí GEM
GEMpiv	- přesnost výsledků dosažených pomocí GEM s pivotizací
NEM	- přesnost výsledků dosažených pomocí NEM
NEMpiv	- přesnost výsledků dosažených pomocí NEM s pivotizací

První tabulka u každého z testů obsahuje hodnoty my_eps1 vypočtené pomocí (6.2), v druhé tabulce naleznete hodnoty my_eps2 vypočtené pomocí (6.3). Pro lepší názornost je každá tabulka vynesena do grafu. Grafy jsou použity pro ukázku chyby a mohou obsahovat nepřesnosti.

6.3.1 Testování pro Pascalovu matici

n	$cond(A)$	GEM	GEMpiv	NEM	NEMpiv
6	1.11e+005	3.55e-016	1.10e-013	8.89e-016	9.78e-016
9	2.91e+008	1.20e-015	1.23e-011	3.37e-014	3.75e-014
12	8.76e+011	2.12e-014	3.67e-007	2.15e-013	1.09e-010
15	2.84e+015	5.71e-013	9.84e-004	2.92e-013	2.72e-010
18	1.02e+019	1.64e-012	4.12e-001	2.31e-012	3.71e-006
21	8.37e+021	1.19e-011	1.00e+000	8.86e-012	1.55e-003

Tabulka 6.1: Výsledky testu my_eps1 pro Pascalovu matici

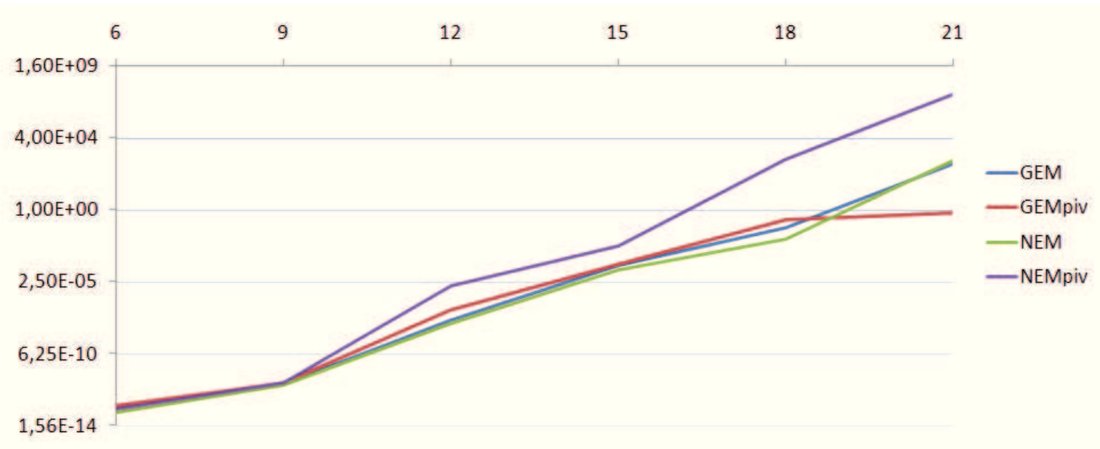


Obrázek 6.1: Chyba my_eps1 pro Pascalovu matici

V tabulce 6.1 vidíme, že podmíněnost matice je vysoká, od rozměru $n = 18$ výrazně přesahuje hranici obrácené hodnoty strojového epsilon. Z tabulky lze vyčíst, že nejpřesnějších výsledků jsme dosáhli pro NEM a se zanedbatelnou nepřesností jej následuje GEM. Rozmezí jejich přesnosti je 10^{-16} až 10^{-11}. S podstatným rozdílem v přesnosti, zhruba 10^{-16} až 1, jsme získali výsledky pro NEMpiv a GEMpiv, přičemž NEMpiv je nepatrně přesnější.

n	$cond(A)$	GEM	GEMpiv	NEM	NEMpiv
6	1.11e+005	2.52e-013	2.84e-013	1.04e-013	1.72e-013
9	2.91e+008	7.39e-012	8.14e-012	5.77e-012	8.77e-012
12	8.76e+011	8.50e-008	3.59e-007	5.71e-008	1.25e-005
15	2.84e+015	2.95e-004	3.47e-004	1.48e-004	4.57e-003
18	1.02e+019	6.68e-002	2.35e-001	1.44e-002	1.74e+003
21	8.37e+021	8.37e+002	6.31e-001	1.52e+003	2.43e+008

Tabulka 6.2: Výsledky testu my_eps2 pro Pascalovu matici



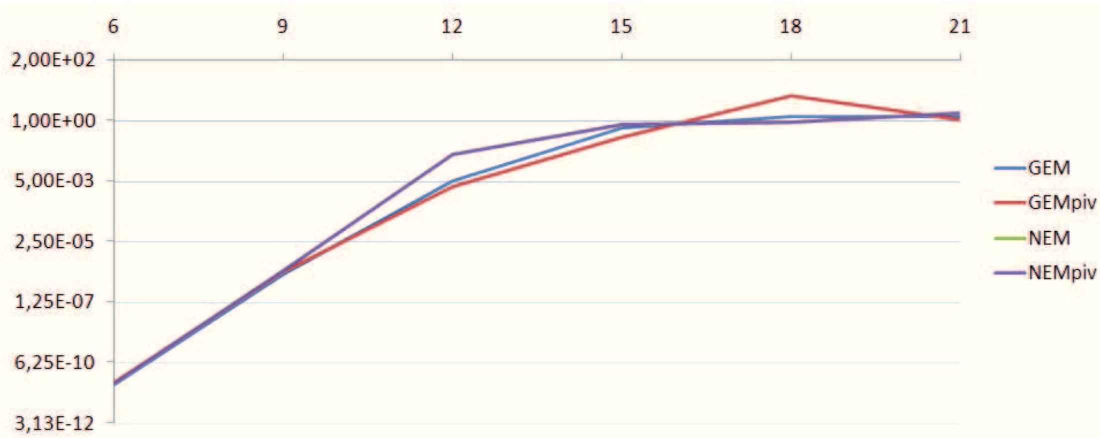
Obrázek 6.2: Chyba my_eps2 pro Pascalovu matici

Z tabulky 6.2 je zřejmé, že nejpřesnějších výsledků jsme dosáhli pro GEMpiv. Druhý nejpřesnější výsledek jsme získali pro GEM a se zanedbatelnou odchylkou jej následuje NEM. Nejhorší výsledky jsme dostali pro NEMpiv. Zde se chyba v závislosti na řádu n velmi zvyšuje, jedná se zhruba o rozsah 10^{-13} až 10^{-3} pro matice s přesně vyjádřitelným číslem podmíněnosti $cond(A)$.

6.3.2 Testování pro Hilbertovu matici

n	$cond(A)$	GEM	GEMpiv	NEM	NEMpiv
6	1.50e+007	9.29e-011	1.10e-010	1.05e-010	1.05e-010
9	4.93e+011	1.47e-006	1.83e-006	2.04e-006	2.04e-006
12	1.68e+016	5.31e-003	3.07e-003	5.57e-002	5.57e-002
15	4.43e+017	5.23e-001	2.48e-001	7.45e-001	7.45e-001
18	6.33e+017	1.50e+000	9.34e+000	8.76e-001	8.76e-001
21	5.10e+018	1.43e+000	1.10e+000	2.01e+000	2.01e+000

Tabulka 6.3: Výsledky testu my_eps1 pro Hilbertovu matici

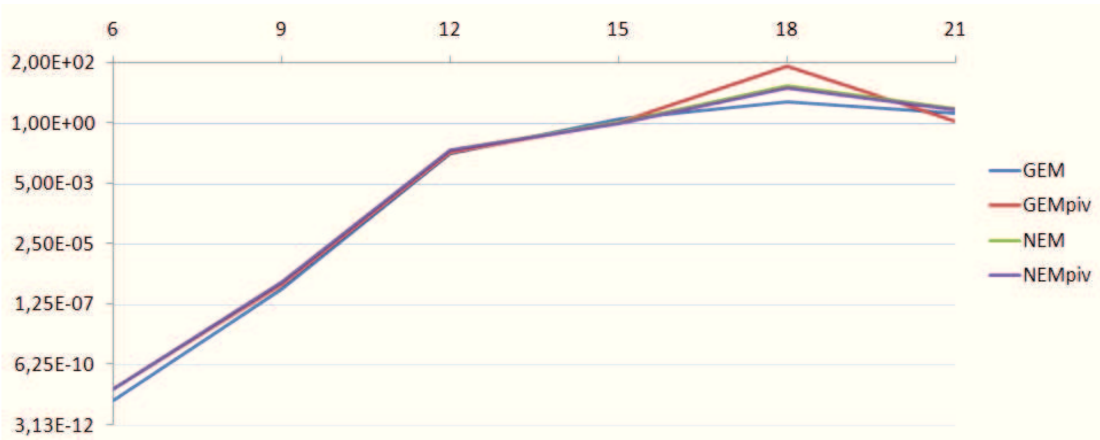


Obrázek 6.3: Chyba my_eps1 pro Hilbertovu matici

Pro Hilbertovu matici jsme u všech algoritmů získali téměř shodnou přesnost. Nepatrně přesnější jsou GEM a GEMpiv, následují NEM a NEMpiv. Přesnost se zvětšujícím se n výrazně klesá, jde o rozdíl v řádu 10^{-10}.

n	$cond(A)$	GEM	GEMpiv	NEM	NEMpiv
6	1.50e+007	2.70e-011	7.71e-011	7.48e-011	7.48e-011
9	4.93e+011	4.73e-007	7.52e-007	9.04e-007	9.04e-007
12	1.68e+016	7.44e-002	7.81e-002	9.11e-002	9.11e-002
15	4.43e+017	1.47e+000	1.09e+000	1.08e+000	8.66e-001
18	6.33e+017	6.78e+000	1.56e+002	2.69e+001	2.13e+001
21	5.10e+018	2.43e+000	1.20e+000	3.65e+000	3.02e+000

Tabulka 6.4: Výsledky testu my_eps2 pro Hilbertovu matici



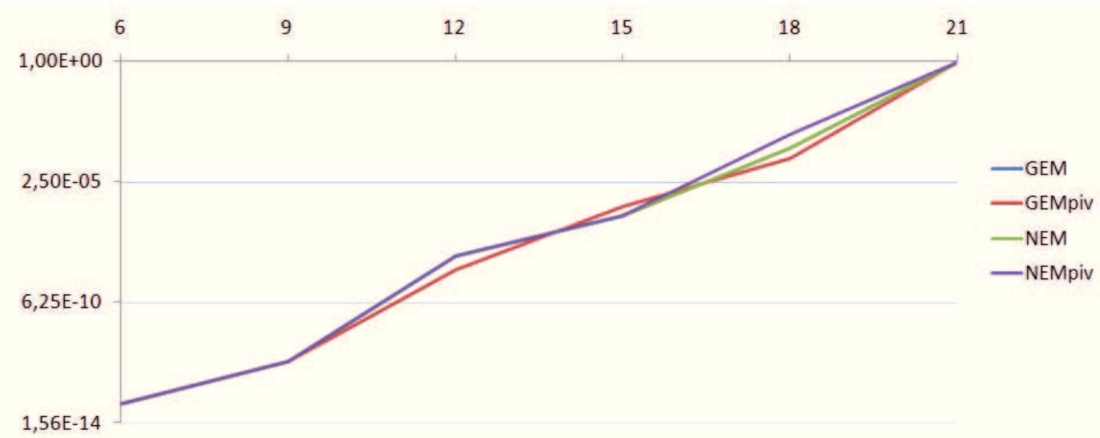
Obrázek 6.4: Chyba my_eps2 pro Hilbertovu matici

V tabulce 6.4 vidíme, že přesnost se vzrůstajícím n klesá. Rozmezí přesnosti je přibližně 10^{-11}. Pro konkrétní n jsou výsledky testu téměř stejné. Nejpřesnější jsou NEMpiv a NEM, se zanedbatelnou chybu je následují GEMpiv a GEM.

6.3.3 Testování pro Vandermondovu matici

n	$cond(A)$	GEM	GEMpiv	NEM	NEMpiv
6	7.31e+005	8.42e-014	8.39e-014	8.49e-014	8.55e-014
9	4.23e+010	3.60e-012	3.57e-012	3.60e-012	3.72e-012
12	6.98e+015	3.87e-008	1.14e-008	3.87e-008	3.86e-008
15	9.75e+019	1.29e-006	2.99e-006	1.29e-006	1.33e-006
18	2.31e+023	4.73e-004	1.93e-004	4.72e-004	1.64e-003
21	2.22e+028	9.31e-001	9.43e-001	9.31e-001	9.30e-001

Tabulka 6.5: Výsledky testu my_eps1 pro Vandermondovu matici

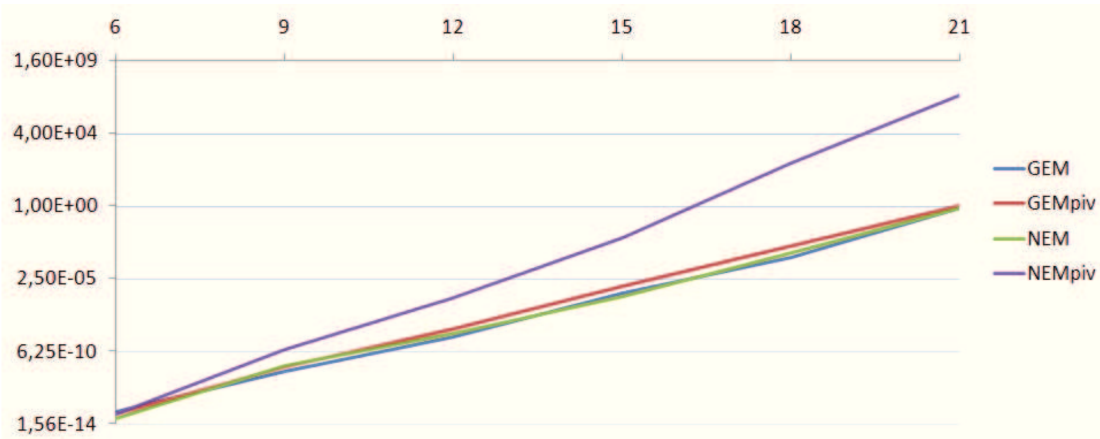


Obrázek 6.5: Chyba my_eps1 pro Vandermondovu matici

S Vandermondovou maticí jsme dosáhli pro určité n přibližně stejné přesnosti pro všechny algoritmy. V tomto případě je těžké určit, který z algoritmů je přesnější. Pro $n = 15$ je nejpřesnější GEM a NEM, poté NEMpiv a nejhorší GEMpiv. Chyba se s rostoucím n výrazně zvyšuje. Přesnost je v rozmezí 10^{-14} až 10^{-1}.

n	$cond(A)$	GEM	GEMpiv	NEM	NEMpiv
6	7.31e+005	8.60e-014	6.12e-014	3.33e-014	6.40e-014
9	4.23e+010	3.19e-011	6.75e-011	7.56e-011	7.31e-010
12	6.98e+015	4.42e-009	1.75e-008	8.39e-009	1.49e-006
15	9.75e+019	2.79e-006	8.40e-006	1.68e-006	1.04e-002
18	2.31e+023	5.41e-004	3.01e-003	9.99e-004	5.86e+002
21	2.22e+028	7.19e-001	1.14e+000	7.32e-001	1.12e+007

Tabulka 6.6: Výsledky testu č.2 pro Vandermondovu matici



Obrázek 6.6: Chyba my_eps2 pro Vandermondovu matici

Z tabulky (6.6) vyplývá, že nejpřesnější výsledek jsme získali pro GEM a NEM. Trochu větší chybu vykazuje GEMpiv a nejhorší přesnost má NEMpiv. Chyba se pohybuje v rozmezí 10^{-14} až 10^{-2} pro matice s vyjádřitelným číslem podmíněnosti.

Kapitola 7

Závěr

Cílem této práce bylo implementovat algoritmy pro řešení soustavy lineárních rovnic GEM, GEM s pivotizací, NEM a NEM s pivotizací v programovacím prostředí MatLab a porovnání přesnosti výpočtu pomocí těchto metod.

Testování přesnosti výpočetních algoritmů probíhalo na soustavě lineárních rovnic s maticí se známou inverzí - konkrétně s Pascalovou, Hilbertovou a Vandermondovou maticí, což jsou čtvercové, regulární, symetrické a totálně pozitivní matice. Bohužel jsou také tyto matice špatně podmíněny, což výrazně ovlivňovalo přesnost získaných výsledků.

Očekávala jsem, že nejpřesnější algoritmus bude NEM s pivotizací a GEM s pivotizací bude zhruba se stejnou chybou. Následovat by je měl algoritmus NEM a nejhorší by měl být algoritmus GEM. Tento výsledek jsem předpokládala podle výsledků z práce [5].

Ze získaných výsledků lze říci, že pro Pascalovu matici byl nejpřesnější algoritmus GEM a NEM, následoval GEM s pivotizací a nejhorší byl NEM s pivotizací. Pro Hilbertovu matici jsem došla k přibližně stejným výsledkům, kde nepatrně přesnější byly NEM a NEM s pivotizací. Pro Vandermondovu matici byl nejpřesnějším algoritmem GEM a poté NEM. S výrazně větší chybou následovaly GEM s pivotizací a NEM s pivotizací.
Když tyto výsledky shrnu, tak nejpřesnějším algoritmem vyšel GEM a NEM, další v pořadí byl GEM s pivotizací a nejhorší byl NEM s pivotizací. Získané výsledky překvapivě neodpovídají očekávání.

Literatura

[1] DOSTÁL Z.,VONDRÁK V., *Lineární algebra*: skriptum. Verze 24.4.2012, Vysoká škola Báňská - Technická univerzita Ostrava a Západočeská univerzita v Plzni, 2012. Citováno 10.4.2013
`http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/linearni_algebra.pdf`

[2] KOZUBEK T., BRZOBOHATÝ T., JAROŠOVÁ M., HAPLA V., MARKOPOULOS A., *Lineární algebra s matlabem*: skriptum. Verze 12.6.2012, Vysoká škola Báňská - Technická univerzita Ostrava a Západočeská univerzita v Plzni, 2012. Citováno 16.4.2013
`http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/linearni_algebra_s_matlabem.pdf`

[3] VONDRÁK V.,POSPÍŠIL L., *Numerické metody I*: skriptum. Verze 2011, Vysoká škola Báňská - Technická univerzita Ostrava a Západočeská univerzita v Plzni, 2012. Citováno 16.4.2013
`http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/numericke_metody.pdf`

[4] KUČERA R., *Numerické metody*: studiijní opora. Vysoká škola Báňská - Technická univerzita Ostrava. Citováno 20.4.2013
`http://homel.vsb.cz/~kuc14/textyNM/kap3.pdf`

[5] ALONSO P., DELGADO J., GALLEGO R., PEÑA J.M., *Growth factors of pivoting strategies associated with Neville elimination*: článek z Journal of Computational and Applied Mathematics 2009. Citováno 28.4.2013

Přílohy

CD příloha obsahuje soubory:

TabHilb.txt - tabulky pro testování Hilbertovy matice

TabPascal.txt - tabulky pro testování Pascalovy matice

TabVander.txt - tabulky pro testování Vandermondovy matice

Dále obsahuje MatLabovské soubory:

GaussElim.m - implementace Gaussovy eliminační metody

GaussElimPiv.m - implementace Gaussovy eliminační metody s pivotizací

NevilleElim.m - implementace Nevillovy eliminační metody

NevilleElimPiv.m - implementace Nevillovy eliminační metody s pivotizací

HilbTab.m - metoda zapisující výsledky testů Hilbertovy matice do *TabHilb.txt*

PascalTab.m - metoda zapisující výsledky testů Pascalovy matice do *TabPascal.txt*

Vander.m - metoda vracející Vandermondovu matici

VanderTab.m - metoda zapisující výsledky testů Vandermondovy matice do *TabVander.txt*